MERKLE PATRICIA TRIE

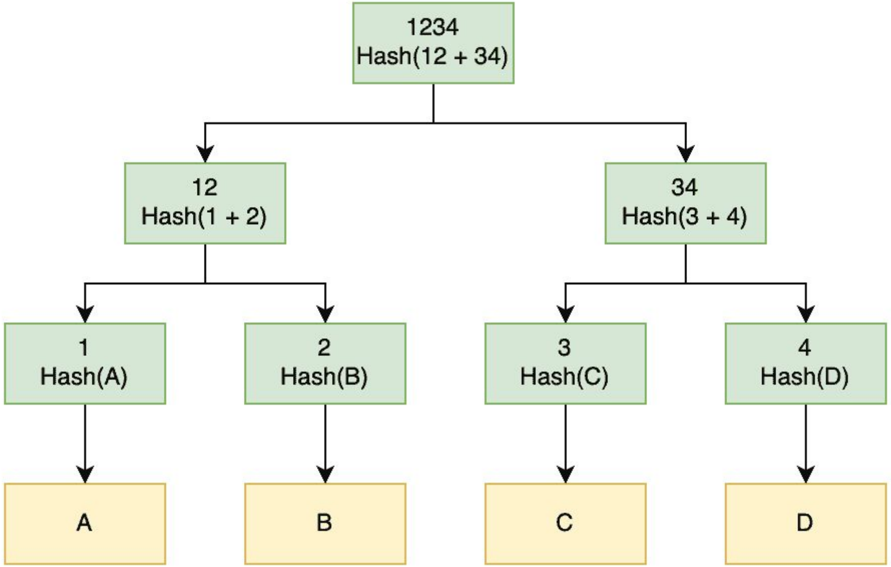
Efficient data retrieval and verification

CONTENTS

- Merkle Tree
- Merits of Merkle Tree
- PATRICIA Trie
- Merits of PATRICIA Trie
- Merkle PATRICIA Trie
- Merits of Merkle PATRICIA Trie
- Code and References

MERKLE TREE

A Merkle Tree is a tree of hash values in which every intermediate node is a hash of its child nodes and the leaf nodes are the hashes of the original data.

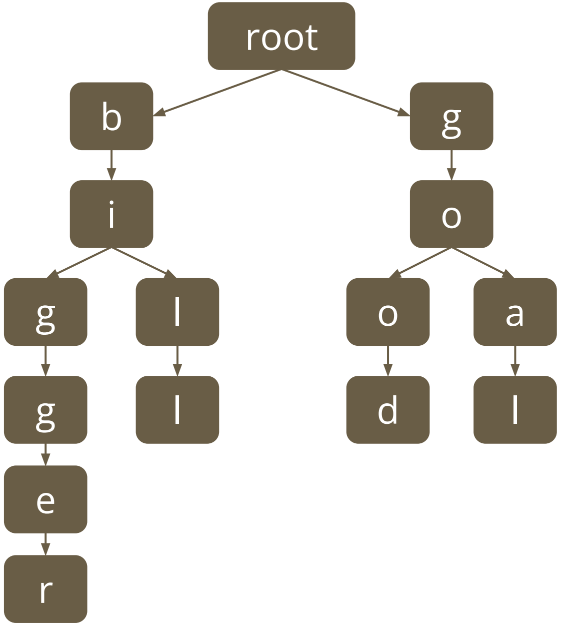


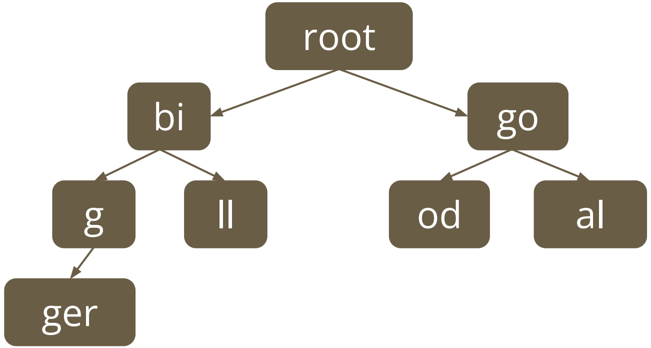
Merits of Merkle Tree

- Size of each node is fixed, as hashes are of fixed sizes
- Data Integrity of large quantities of data can be easily verified by just comparing the root hashes in O(1) time
- If we have a trusted source for root hash, we can obtain the entire tree from an untrusted source and still prove that it’s correct
- Presence of data can be proved in O(logN) time

PATRICIA Trie

A PATRICIA (the Practical Algorithm To Retrieve Information Coded in Alphanumeric) Trie is a data structure that represents a space-optimized prefix tree in which each node that is the only child is merged with its parent.





<u>Words stored in above Tries:</u>

“big”, “bigger”, “bill”, “good”, “goal”

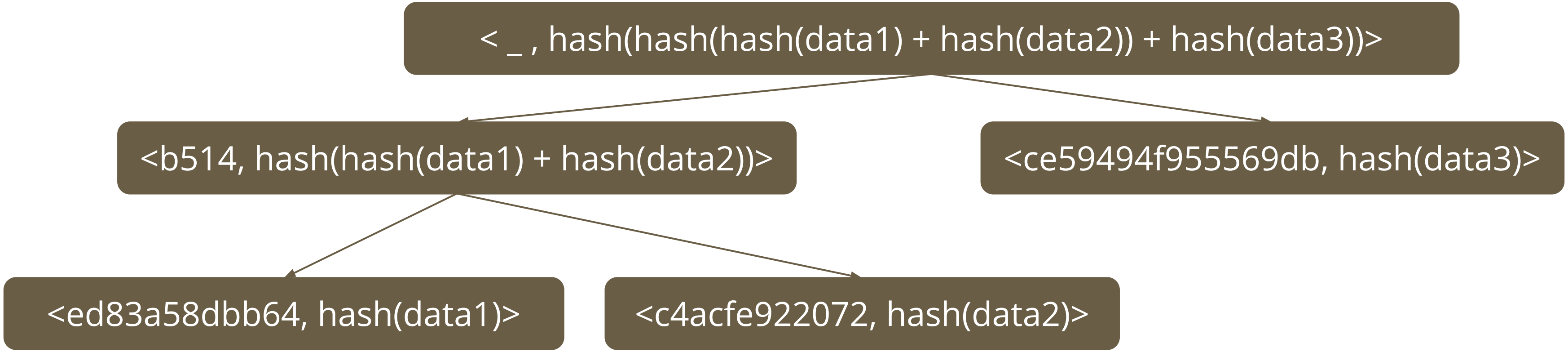
Merits of PATRICIA Trie

- A node’s position in a trie defines the key with which that node is associated. This makes tries more space-optimized than binary search Trees, in which a node stores a key that corresponds only to that node.

- Patricia tries are the most compressed versions of corresponding tries, so they save even more space

MERKLE PATRICIA TRIE

A Merkle Patricia Trie is a patricia trie in which every intermediate node is a hash of its child nodes and the leaf nodes are the hashes of the original data. Thus, it introduces the core feature of merkle trees into patricia tries.



< _ , hash(hash(hash(data1) + hash(data2)) + hash(data3))>
<b514, hash(hash(data1) + hash(data2))>
<ce59494f955569db, hash(data3)>
<ed83a58dbb64, hash(data1)>
<c4acfe922072, hash(data2)>

Data represented by above Merkle Patricia Trie:

[[“b514ed83a58dbb64”, data1], [“b514c4acfe922072”, data2], [“ce59494f955569db”, data3],]

Merits of Merkle Patricia Trie

- Size of each node is fixed, as hashes are of fixed sizes
- Data Integrity of large quantities of data can be easily verified by just comparing the root hashes in O(1) time
- Presence of data can be proved in O(logN) time
- Much more space optimized in comparison to trees and general tries

CODE AND REFERENCES

- My Implementation: https://github.com/DevRish/merkle-patricia-trie
- https://ethereum.org/en/developers/docs/data-structures-and-encoding/patricia-merkle-trie/
- https://www.researchgate.net/publication/358740207_An_Overview_of_Trees_in_Blockchain_Technology_Merkle_Trees_and_Merkle_Patricia_Tries
- https://en.wikipedia.org/wiki/Radix_tree
- https://en.wikipedia.org/wiki/Merkle_tree